

Conventional DSP or configurable microcontroller: which way to go?

ENGINEERS OF DSP-BASED SYSTEMS ARE IN A QUANDARY OVER WHETHER TO USE

A CONVENTIONAL DSP OR ONE OF THE NEW CONFIGURABLE MICROCONTROLLERS.

THE TECHNICAL CHALLENGES OF MP3 DECODING PROVIDE THE BASIS FOR COMPARING

TRADITIONAL DSPs WITH UPSTART CONFIGURABLE MICROS.



At a glance.....66

ARC instruction set.....68

THE TECHNOLOGY LANDSCAPE for digital-signal processing is rapidly changing. DSP performance continues to increase at a fairly constant rate across the industry, and the complexity of peripherals is also expanding. But whereas DSP-centric applications have traditionally employed processors designed for DSP tasks, a

complementary, and some say competing, technology has emerged. Nearly every microcontroller supplier has jumped on the DSP bandwagon, offering DSP-specific additions to their architectures. Some of these microcontrollers have a multiply-accumulator mapped into the address space, and others integrate DSP-specific instructions into the instruction set. Along with these changes comes a whole raft of tools proposed to measure a product's fitness as a DSP, a microcontroller, or an application-specific device. Benchmark code provides one type of normalized performance result. Companies, such as BDTi, employ systems that combine a suite of benchmark code that manually optimizes BDTi for specific processors. When engineers combine the benchmark results with a narrative report on coding strengths and weaknesses, they gain a view into the probable workings of the processor. But most of these measurements are missing the effect of having the right selection of peripherals available to the engineer.

A wide selection of peripherals has been the hallmark of microcontrollers for a long time. You can find microcontrollers that come close to providing the perfect selection of performance, memory, and peripherals for many applications. LCD drivers, high-voltage interfaces for high-side drivers, A/D converters, serial interfaces, and a whole series of communications-protocol peripherals are just the tip of the iceberg for microcontroller peripherals. But the selection is narrower for DSPs. Most DSP-product families have a small offering of peripherals, and most of these peripherals are dedicated to basic I/O functions and serial-communications protocols. Part of the reason for the wider selection of peripherals available on standard off-the-shelf microcontrollers comes from the long history of microcontroller pervasiveness in consumer goods. But the more significant reason for the greater selection of standard-configured microcontrollers is the intense competition for every application that a microcontroller can implement. The result of this com-

petition is the configurable ASIC-based microcontroller. ARC cores offers the prototypical ASIC-based extensible and configurable microcontroller core. Based on a 32-bit instruction set, the ARC architecture allows engineers to configure and extend the capabilities of the processor, including custom instructions and unique peripherals.

Unlike traditional embedded processors that companies originally intended as discrete components for pc-board-based applications, ARC designed its core for SOC (system-on-chip) designs. As a result, you can integrate an ARC core into an SOC much faster than you can with other processors, often in just a few days. ARC delivers its cores as soft macros. This method eliminates dependence on a specific foundry, enabling the selection of the most cost-effective vendor or permitting second sourcing. It's also easy to quickly take advantage of new manufacturing technologies with simple resynthesis instead of the lengthy and expensive process required to port a hard macro or wait for a standard product to become available in the new process.

A comprehensive set of development tools for both software and hardware development fully supports ARC processors. These tools include the ARChitect, a graphical environment that enables "point-and-click" configuration of the processor in minutes. You need not manually modify HDL code; the tools automatically handle all of the modifications based on choices you make in the graphical environment.

Many current designs use both a microprocessor core and a DSP core. ARC cores permit you to use one architecture for both requirements. ARC allows engineers to either combine both functions onto one core or configure separate ARC processors, one as a microcontroller and one as a DSP.

DSPs DO THE JOB

Compared with configurable microcontrollers, Analog Devices' standard-product fixed-point DSPs provide capabilities that are typical of many DSPs. Apart from the long-standing architecture

AT A GLANCE

- ▷ Microcontrollers, unlike most DSPs, have a wide selection of peripherals due to competition for microcontroller applications.
- ▷ DSPs meet the diverse needs of many applications at an attractive price.
- ▷ Your engineering requirements determine whether a DSP or a microcontroller is right for you.

that supports C-like assembly language, the ADSP-2186M is notable for its low price and pin-compatible family road map. The ADSP-2186M combines the ADSP-2100-family base architecture with two serial ports, a 16-bit internal DMA (direct-memory-access) port, a byte-wide DMA port, a programmable timer, flag I/O, and interrupt capabilities. The part can sustain a 13-nsec instruction-cycle time (75 MIPS) while operating on a 2.5V supply with 3.3V I/O. Memory consists of 8k words of on-chip program RAM and 8k words of on-chip data-memory RAM. Like many microcontrollers, the ADSP-2186M includes features aimed at simplifying systems design, including programmable memory strobe and separate I/O memory space.

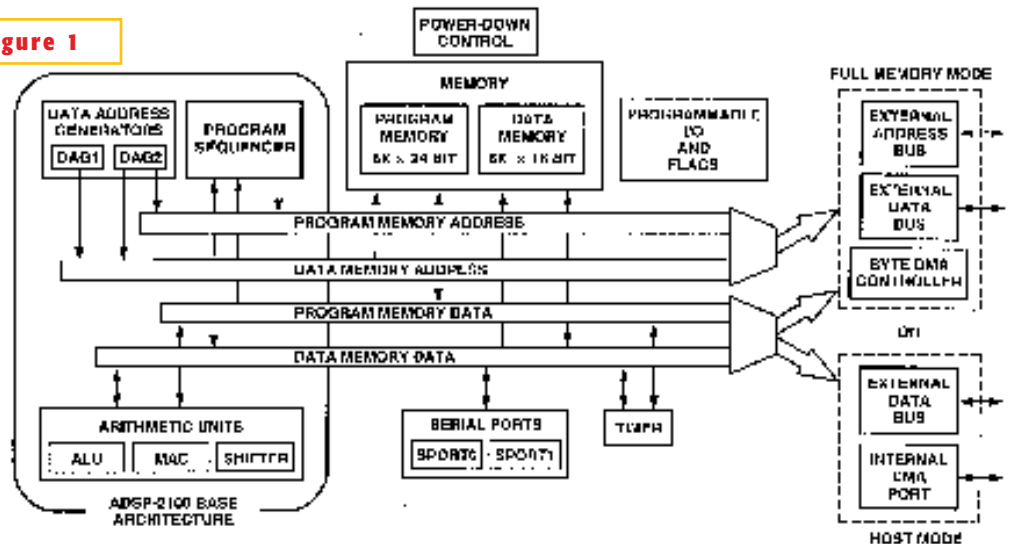
MP3 has gained notoriety as a result of people using the Internet to exchange

high-quality copies of songs. MP3 compresses audio to a fraction of the full CD bit stream, making practical the exchange of songs over the Internet. The decoding process requires modest DSP computational power and minimal interface capabilities. Although the decoding requirements are modest by DSP standards, they require significant processing by microcontroller standards. The decoding process includes bit-oriented and control-intensive algorithms. In a very real sense, MP3 decoding is a hybrid algorithm requiring the best of both DSPs and microcontrollers.

MPEG bit streams divide into packets called "frames." Every MPEG format includes a fixed number of frames per second. For a specific bit rate and sampling frequency, each frame has a fixed size and produces a fixed number of output samples. MPEG-audio frames in the decoding process are independent of each other. This independence means that it is simple to skip forward or backward in a bit stream, because you can simply skip forward or backward to the next or previous frame. The first step in the decoding process is finding the beginning of a frame by searching for a synchronization bit pattern. After you find it, you can read the frame header and side information to learn how the frame was encoded to enable the decoder to correctly process its data.

The first data transmitted in the data

Figure 1



The Analog Devices DSP includes the hallmarks of traditional DSP products: a specialized multiply-accumulate unit; parallel addressing units with mathematical capabilities; and other mathematical hardware, such as a barrel shifter. Units are arranged in parallel to permit simultaneous operation.

stream after the header are the scale factors. These scale factors control the gain for specific frequency bands. Next in the data stream are the actual frequency energies, which are quantized and Huffman-encoded. The decoder's task then is to decode the Huffman codes, requantize the results, and transform the energies into the time domain so that the output stream accurately represents the original signals. The Huffman encoding in the encoder minimizes total bit length by employing many Huffman trees, which the data contents determine. The Huffman decoder selects the appropriate Huffman tree and then traverses the tree for each energy symbol in the frame data to arrive at the decoded value. When the Huffman

decoder has decoded the values, the quantizer rescales them using the scale factors to re-create the real spectral-energy values.

Bit streams are conceptually simple but can become more complicated as you increase the number of channels. The simplest structure for a stereo signal is for it to transmit each channel separately in every frame. More capable encoders exploit redundancies between the two channels by transmitting the sum and the difference to further compress the original signals. When the signal encoding employs this type of representation, the decoder must perform stereo processing to recover the original two channels. Next, the decoder symmetrically adds the

frequency values in each frequency band to smooth out aliasing distortions that quantizing causes.

IDCT IMPLICATIONS

So far, this discussion has focused on signals in the frequency domain. To synthesize the output samples, you apply a transform that is the reverse of the time-to-frequency transform that the encoder uses. In Layer 3 encoding, two transforms are applied after each other by the IDCT (inverse-discrete-cosine-transform) stage to achieve better frequency resolution than in the other layers. Both transforms are critically sampled DCTs, meaning that if no quantizing occurs during encoding, the decoder perfectly

ARC INSTRUCTION SET

ARC's Tangent-core processor is unique among microcontrollers, but companies are starting to move their microcontroller offerings in Tangent's direction. The ARC core provides 30 base instructions, and you may configure it to include as many as 71 additional user-defined instruction codes.

In addition to the base-instruction set, the ARC processor includes predefined extension instructions that you can use to accelerate algorithms in your applications. The extensions libraries include a powerful set of DSP instructions and additional general-purpose microcontroller instructions, such as barrel shifters. In addition to predefined instruction extensions, you can easily add custom instructions using VHDL or Verilog. You can easily configure the ARC software tools to enable use of these instructions in the standard software-development flow.

The ARC processor uses a four-stage pipeline that supports both single-cycle and multicycle instructions, enabling you to add highly complex instructions to the pipeline. The ARC processor provides delay-slot execution modes that support zero-overhead loops and branches. All

arithmetic-logic-unit instructions are conditional. There are 32 possible condition codes: 16 codes are predefined, and 16 codes are available for user-defined extensions. This situation enables you to add custom condition codes that peripheral hardware, such as custom coprocessors, can set. These custom codes enable application-specific software to implement complex decision-making algorithms in a highly efficient manner, as both peripherals and instruction extensions can create condition codes for particular decisions.

REGISTERS

The ARC core provides 32 general-purpose, 32-bit-core registers. You can increase the total number of registers to as many as 64, to include any user-defined core registers. It also provides a 4-Gbit auxiliary register space that you access via the auxiliary bus. This access enables you to define an essentially unlimited number of auxiliary registers to interface other peripherals.

DSP EXTENSIONS

The ARC processor offers a powerful set of DSP extensions that deliver real DSP perfor-

mance, including XY memory, a configurable MAC, and saturating arithmetic operations.

The configurable MAC supports 16×16 , 24×24 , and dual 16×16 modes, enabling you to select the appropriate precision for your application. The configurable XY memory system allows as many as four banks of XY memory. It enables sustained performance, because the processor can access one bank while a fast DMA transfer or other memory access is occurring in another bank. Four 32-bit registers associated with each memory bank provide fast context switching. To accelerate functions, such as FFTs, address-generation units for XY memory support modulo and bit-reverse addressing with variable-offset pre-increment and postincrement modes. To reduce area requirements, you can remove these address modes during processor configuration.

INSTRUCTION AND DATA CACHES

ARC processors provide instruction and data caches that greatly improve load-store and instruction-fetching efficiency.

Both instruction and data caches are completely configurable. You can configure the cache size and choose one-,

two-, or four-way set associativity. The line length is configurable, and you can lock lines, ensuring that performance-critical data or code, such as interrupt-service routines, is always cached. You can select from two runtime replacement algorithms: random replacement or round-robin replacement. And, you can invalidate the entire cache or just individual cache lines.

The data cache provides write-back, blocking, and write-allocate write policies; a one-line flush buffer; and cache flush under CPU control. LD.DI and ST.DI instructions provide a bypass mode for volatile data.

POWER MANAGEMENT

The ARC processor's small size and configurability enable you to design low-power options by eliminating functions that your application doesn't use. The ARC processor also provides a sleep mode and clock-gating options that enable you to control when the processor powers up and which execution units are enabled. You can dramatically reduce power consumption through the use of custom or extension instructions that achieve algorithms' performance targets at much lower clock speeds.

reconstructs the original signal. To avoid discontinuities between transformed blocks, which would result in audible noise and clicks, the transforms use a 50% overlap. So, for a block size of X, the encoder advances the input pointer only X/2 samples for each transformed block. The decoder does the reverse; every re-transformed block sample overlaps with half of the previous block's samples. This process smoothes out any discontinuities. After the transform occurs in the decoder, a lowpass filter is applied to the results to produce output signals. The decoder implements the lowpass filter by convolution with a sinc-shaped wave. Technically, every output sample becomes a weighted average of the surrounding 512 samples that you have transformed in the time domain, and the MPEG-audio standard predefines the shape of the 512 weights.

You can define the IDCT by

$$s(y, x) = \sum_{u=0}^7 \sum_{v=0}^7 \alpha(u)\alpha(v)f(v, u) \cos \frac{(2x+1)\pi u}{16} \cos \frac{(2y+1)\pi v}{16},$$

where you define the normalization factors as

$$\alpha(u) = \frac{1}{2\sqrt{2}} = \frac{1}{\sqrt{8}} \text{ for } u=0, = 1/2 \text{ for } u > 0.$$

The summation requires 4096 multiplications and 4032 additions when you implement it strictly according to the form of the equation. Implementing this algorithm in a straightforward manner requires about 3 million multiply-accumulate instructions per second. Judicious code optimization can reduce this number by a factor of two. You can implement the MP3 algorithm in approximately 15 DSP MIPS—well within the capabilities of nearly every available DSP. The nature of the IDCT algorithm allows you to simply evaluate the two architectures for DSP functionality.

DSP FOR IDCT

Figure 1 illustrates the architecture of the ADSP-2186M. The single-cycle non-pipelined design simplifies technical analysis, because there are no delays or other issues normally associated with pipelines. The 2186 includes a single-cycle multiply-accumulate instruction that can independently and simultaneously address two operands. Unfortunately, the

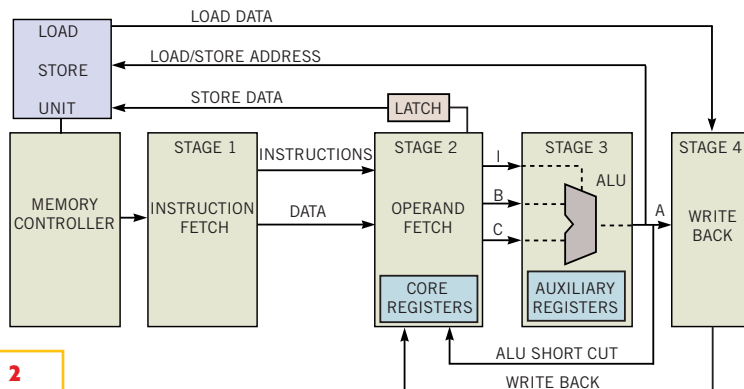


Figure 2

ARC's Tangent processor core relies on pipelining to permit faster normal operation. By employing conservative design practices coupled with a pipeline operation, the Tangent core can maintain competitive performance. Unlike the ADO DSP, the Tangent's memory space is a single-access linear memory.

specialized zigzag-addressing capabilities that form the backbone of the IDCT algorithm do not fit the modulo-addressing capabilities of the 2100 family. This problem is not a big one, however, because most of the code optimizations employed in speeding the IDCT algorithm use fully loop-unrolled code. This code does not rely directly on the addressing mechanism; instead, it uses hard-coded addresses. A listing on EDN's Web site shows that the multiply-accumulate operation can directly perform most operations; find it at www.ednmag.com/ednmag/reg/2001/01042001/01df1.htm. Optimizing the algorithm for the 2186 is a well-understood process that simply takes time and cleverness.

In contrast, ARC's Tangent core is a pipelined CPU employing a RISC-based load-store architecture (**Figure 2**). (See sidebar "ARC instruction set.") The core employs an instruction pipeline, which is a sequence of functional units that performs an instruction in several steps. Each functional unit accepts inputs and creates outputs that its output buffer stores. In the pipeline, the output of one stage becomes the input of the next stage, allowing all the stages to work in parallel. This setup results in greater throughput. ARC built its Tangent core around a pipeline, with different stages performing tasks such as fetch instruction, decode instruction, load data, and store results, as well as arithmetic operations. The instruction pipeline relies on a continuous stream of instructions that it fetches from sequential locations in memory. This stream is interrupted

when a branch is taken. The contents of early stages contain instructions that should not execute from locations after the branch instruction; this condition is known as a pipeline break.

Implementing the unoptimized MP3 IDCT algorithm using the basic ARC core requires 4096 multiplications, 4032 additions, 8128 address computations, and a similar number of operand fetches. Although this rate is more than six times the MIPS rate that the 2186 requires, or about 100 Tangent MIPS, it's well within the capabilities of the Tangent-core processor. But the issue is a bit more complex. Because the Tangent core employs a multistage pipeline, you must consider issues of pipeline stalls and breaks when evaluating the performance that achieving real-time operation requires. Pipeline stalls happen when required data is unavailable when the processor needs it. The processor typically stalls when it does not obtain a result through the calculation process or when memory-access times are too long. Breaks occur when program flow takes a branch, such that the target instruction is outside the pipeline. When you use the basic core, both pipeline breaks and stalls may exist. But this situation need not occur. You can reorganize the code to avoid stalls, but the increased latency due to breaks remains.

POWER BUDGETS APPLY

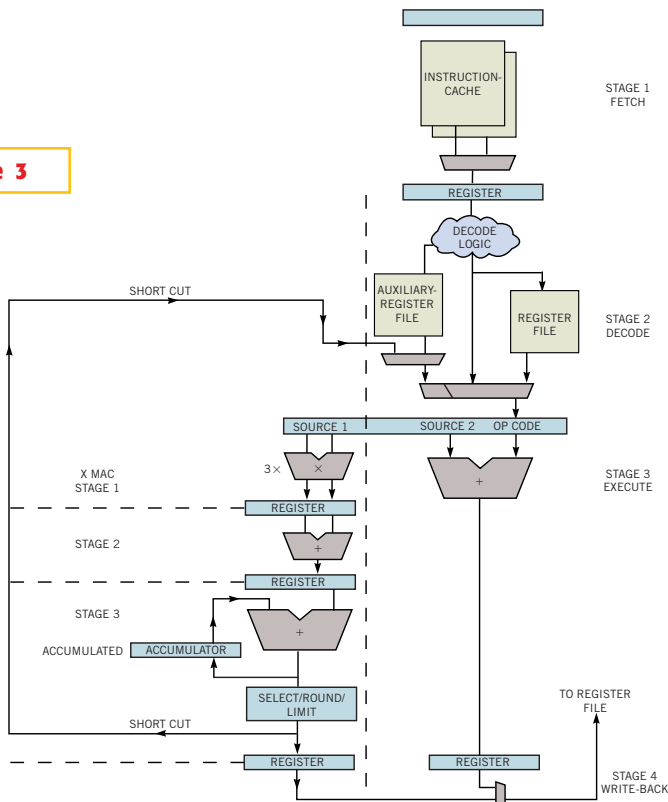
Assuming that both processors consume similar amounts of power per MIPS, Analog Devices' DSP would win the power-budget battle. But, Tangent

permits engineers to choose a multiply-accumulate instruction in addition to the instruction set, a dual-ported XY memory, and full DSP modulo addressing. So, except for the load-store architecture, the Tangent processor can achieve the same MIPS rate as modern main-stream DSPs. Consequently, the two processors consume the same amount of power when you implement them in the same process. But ARC goes all standard DSPs one better. Because the company delivers the processor in a soft-macro form, you can target low-power processes and still yield the MIPS rating that the application requires. In this case, the configurable core has an important advantage, but this advantage may come with some penalties. For a hand-optimized DSP manufactured in a standard semiconductor process, the die size for the finished part will likely be smaller than the equivalent synthesized part using the ARC core. The size differential depends on many factors, including the amount and type of on-chip memory, the maturity and robustness of the target design library, the quality of the synthesis tools, and the skill of the engineer controlling the synthesis process. Traditionally, the amount and type of memory on the die has the greatest impact on DSP die size. To achieve high density and low power while meeting performance budgets, memories are usually handcrafted. So, the die sizes can be equivalent between standard DSPs and configurable micro-controllers.

GILDING THE LILY

Extensible cores, such as Tangent, allow designers to go one step further than standard DSPs. With MP3 players, you can implement an IDCT instruction. Although the instruction is limited by the number of cycles that it requires to complete a full transform, one advantage to creating an application-specific instruction is a further reduction in power consumption. Because one IDCT instruction avoids decoding at least 4096 instructions, power savings are possible due to the efficiency of the custom instruction. But the Tangent core outshines most standard fixed-point DSPs in supporting programs written in C. Because ARC designed Tangent with C in mind, it includes instructions that are useful for compilers but are of little value to assembly-language programmers. Tangent

Figure 3



ARC adds DSP-like architectural elements to improve performance on DSP algorithms. Dual-access XY memory feeds two data words to the multiply-accumulate unit. Addressing hardware brings DSP-like addressing capabilities to the RISC core.

includes an improved ability to change contexts, save processor states, and address stack frames. Although you can implement the basic MP3 decoder in 2000 to 3000 instructions, as the user-interface features become more complex and capable, the programming issue tilts in favor of using C for the control code.

SOCs

You can easily implement a generic MP3 decoder using either device. Each has adequate processing power, can achieve a minimal power budget, and includes enough memory in a basic configuration to implement the software on-chip. But, if you expand the product requirements to embrace a complete MP3-player definition, a whole new set of interface requirements and software comes into play. Assuming that you need only to drive an LCD to show the songs playing and interface to five or six buttons, you can configure the ADSP-2186M to read the buttons, drive a serially controlled LCD driver, and interface with a stereo codec. In comparison, you may configure

the Tangent core with an on-chip LCD driver and purpose-built button inputs. The stereo analog output can be any interface that fits the size and cost budget.

BETTER ONE, BETTER TWO?

Standard DSP products, such as the ADSP-2186M, meet the diverse needs of many applications at an attractive price. But when you consider the product from a systems standpoint, a growing number of consumer and other applications can benefit from the ability to configure and extend a processor. The approach that meets your specific needs depends on the type of I/O you require, the product volume, the performance, the power, the size, and the second-source capabilities. Regardless of your technical requirements, an option is available to meet your needs. It's just a matter of the amount and type of engineering work that you require. □

Henry Davis is a technology writer and consultant based in Soquel, CA. You can contact him at henry@henry-davis.com.